

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-040463

(43)Date of publication of application : 19.02.1993

(51)Int.Cl. G09G 5/24  
G06F 3/153  
G06F 15/20

(21)Application number : 03-199154 (71)Applicant : HITACHI LTD

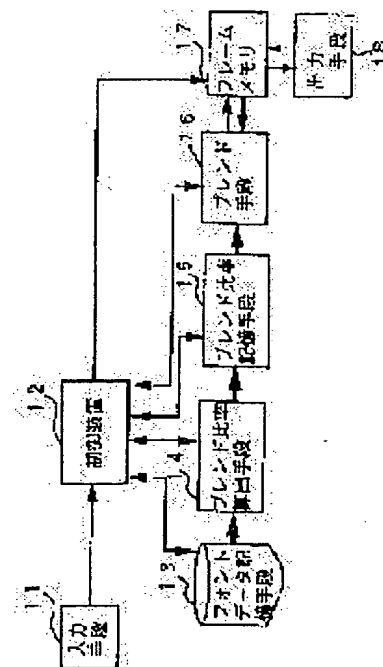
(22)Date of filing : 08.08.1991 (72)Inventor : YAMAZAKI NAOMI  
KAWABATA  
ATSUSHI

## (54) MULTI-LEVEL CHARACTER GENERATOR

### (57)Abstract:

**PURPOSE:** To output a multi-level character in which no burr is prominently observed to the whole display or a printer, etc., at high speed.

**CONSTITUTION:** The above purpose can be attained by providing an input means 11, a font data storage means 13, a blend ratio calculation means 14, a blend ratio storage means 15, a blend means 16, and frame memory 17, and controlling them by a controller 12. In other words, the multi-level character can be generated at high speed and outputted by finding a blend ratio without using working memory.



## LEGAL STATUS

[Date of request for examination] 18.11.1994

[Date of sending the examiner's decision of rejection] 01.07.1997

[Kind of final disposal of application]

other than the examiner's decision  
of rejection or application  
converted registration]

[Date of final disposal for  
application]

[Patent number]

[Date of registration]

[Number of appeal against  
examiner's decision of rejection]

[Date of requesting appeal against  
examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-40463

(43) 公開日 平成5年(1993)2月19日

(51) Int. Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G09G 5/24		9061-5G		
G06F 3/153	310	B 9188-5B		
15/20	562	Z 7343-5L		

審査請求 未請求 請求項の数 6 (全14頁)

(21) 出願番号	特願平3-199154	(71) 出願人	000005108 株式会社日立製作所 東京都千代田区神田駿河台四丁目6番地
(22) 出願日	平成3年(1991)8月8日	(72) 発明者	山崎 直美 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内
		(72) 発明者	川端 敦 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内
		(74) 代理人	弁理士 高田 幸彦

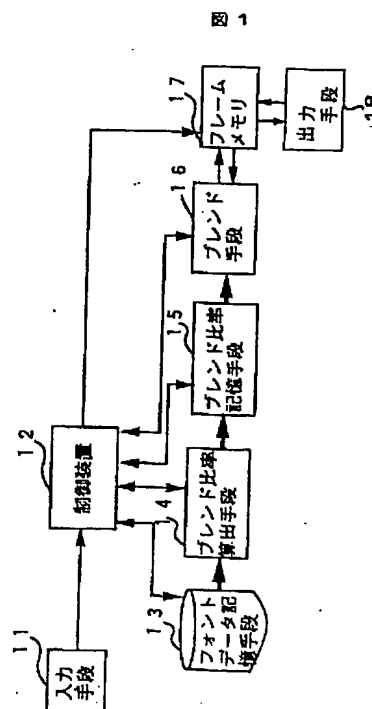
(54) 【発明の名称】 多階調文字発生装置

(57) 【要約】

【目的】 本発明の目的は、ディスプレイ全般やプリンタなどにギザギザが目立たない多階調文字を高速に出力することにある。

【構成】 入力手段(11)フォントデータ記憶手段(13)ブレンド比率算出手段(14)ブレンド比率記憶手段(15)ブレンド手段(16)フレームメモリ(17)を設け、制御装置(12)で制御することによって達成される。

【効果】 ワークメモリを用いずにブレンド比率を求めることにより、多階調文字を高速に生成し、出力することができる。



## 【特許請求の範囲】

【請求項 1】文字コードを表わす情報を入力する入力手段と、

前記文字コードを表わす情報に対応する文字の形状データを記憶するフォントデータ記憶手段と、

前記文字の形状データを構成する各ドット毎に、前記文字の形状データの領域の占める割合を数値計算によって求めるブレンド比率算出手段と、

前記ブレンド比率算出手段によって算出された前記割合に基づいて、各ドット毎に階調情報を生成するブレンド手段と、

前記ブレンド手段により生成された各ドット毎に階調情報を出力する出力手段とを具備することを特徴とする多階調文字発生装置。

【請求項 2】請求項 1 に記載の多階調文字発生装置において、前記ブレンド比率算出手段によって算出された割合を保持するブレンド比率記憶手段を備えることを特徴とする多階調文字発生装置。

【請求項 3】請求項 1、または、請求項 2 において、前記ブレンド比率算出手段は、各ドット単位に文字部分を覆う一方向の短冊状帯領域の面積の総和を計算し、このドットにおける文字領域の占める面積と近似して、前記割合を求める手段であることを特徴とする多階調文字発生装置。

【請求項 4】請求項 1 から請求項 3 の何れかにおいて、文字の存在領域を示すマスクパターンを保持する手段を備えることを特徴とする多階調文字発生装置。

【請求項 5】文字コードを表わす情報を入力する入力手段と、

前記入力手段に接続され、入力された前記文字コードを表わす情報に対応する文字処理コマンドとパラメータとを生成する制御装置と、

前記文字コードを表わす情報に対応する文字の形状データを記憶するフォントデータ記憶手段と、

前記制御手段と前記フォントデータ記憶手段とに接続され、前記文字コードを表わす情報に対応する文字処理コマンドとパラメータと、前記文字コードを表わす情報に対応する文字の形状データとを受取り、前記文字の形状データを構成する各ドット毎の階調情報を生成する文字処理手段と、

前記文字処理手段に接続され、前記文字の形状データを構成する各ドット毎の階調情報を記憶するフレームバッファと、

前記フレームバッファに接続され、前記文字の形状データを構成する各ドット毎の階調情報を出力する出力手段とを具備することを特徴とするデータ処理装置。

【請求項 6】請求項 5 において、上記文字処理手段は、単一の半導体基板に集積化された文字処理手段であることを特徴とするデータ処理装置。

【発明の詳細な説明】

## 【 0 0 0 1 】

【産業上の利用分野】本発明は、パソコンやワークステーションを主としたディスプレイ全般やレーザプリンタ、あるいはカラープリンタなどの多階調表示装置において、多階調文字を表示する多階調文字発生装置に関する。

## 【 0 0 0 2 】

【従来の技術】多階調文字とは、図 8 に示すように、文字の輪郭部分のドットを文字色と背景色との中間色で階調表示したものであり、小さな文字はきれいに、大きな文字も斜線部分のぎざぎざが目立たないように表示できる。

【 0 0 0 3 】従って、多階調文字であれば、描画位置を極くわずかに移動しても前記説明したように文字が変形することはない。

【 0 0 0 4 】ところで多階調文字を生成するには、文字の輪郭部分の各ドットにおいて、ドット面積に対して文字領域の占める割合を求め、背景の色と文字の色を求めた割合に応じて混合し、文字の輪郭部分のドットの色を決定する必要がある。

【 0 0 0 5 】従来は、ベクトル文字情報から文字を構成する各ドットごとのブレンド比率を算出するために以下のような処理を行っていた。

【 0 0 0 6 】まず、文字のベクトルデータを読み込み、展開する。次にあらかじめ膨大な計算用のワークメモリを確保しておき、その上に 1 個のドットを更に縦横に分割したサブピクセルパターンを考え、展開した文字を重ね合わせ、描画する。すなわち図 3 に示すように文字の輪郭線 3 3 内部のサブピクセル 3 2 を塗りつぶす。そして塗りつぶされたサブピクセル 3 2 を数え上げてドットごとに集計し、（塗りつぶされたサブピクセルの数 ÷ 1 ドット内のサブピクセルの総数）より、文字を構成するすべてのドットに対して、文字領域の占める割合（ブレンド比率）を算出する方法が知られていた。これ等に関連する技術として特開昭 63 - 313191 号公報が挙げられる。

## 【 0 0 0 7 】

【発明が解決しようとする課題】このように従来のブレンド比率算出方法では、数倍細かくしたサブピクセル単位のワークメモリ上に文字を一旦描画し、さらに描画後の状態を各ドット単位に調べ上げ、塗りつぶされたサブピクセル数を数える必要があり、処理に時間がかかるという問題があった。このとき図 1 5 に示すように、4 階調の場合には、2 値の 4 倍、1 6 階調の場合には 1 6 倍、6 4 階調の場合には 6 4 倍というように、階調数が増加するにつれて、各ドットのサブピクセル数、すなわち一旦文字を描画するという処理で必要となるワークメモリが面積に比例して増加するため、処理時間も同様に増加し、膨大なワークメモリを確保しなければならず、実用的でなかった。

【0008】上記従来技術では、文字を構成する各ドットの文字領域の占める面積割合を算出する場合に、出力する文字を数倍細かくしたサブピクセル単位のワークメモリ上に一旦描画するため、膨大なワークメモリを確保しなければならないという問題があった。

【0009】また階調数によって、ワークメモリの大きさが大幅に変化するため、実用的でないという問題もあった。

【0010】さらに描画後の状態を各ドット単位に調上げ、塗りつぶされたサブピクセル数を数える必要があり、描画と数え上げにかかる処理時間が長く、文字サイズ及び階調数に依存して処理時間が大幅に変化するという問題も生じていた。

【0011】本発明は、文字を構成する各ドットの文字領域の占める面積割合を算出する場合に、ワークメモリ上に文字を一旦描画して数え上げる処理を一切行わず、数値計算のみの処理で面積割合を算出することにより、膨大なワークメモリを確保する必要がなく、さらに描画及び1ドットずつの数え上げに要していた時間を短縮することができ、高速に多階調文字を生成することを目的としている。

【0012】さらに、文字の存在領域を示すマスクパターンと、ブレンド比率を別に蓄え、ブレンド比率を必要最小限のドット数で管理することにより、必要なメモリ量の削減を目的としている。

【0013】

【課題を解決するための手段】本発明によれば、各ドットごとのブレンド比率を求める際に、従来から高速化の妨げとなっていた、ワークメモリ上に文字を描画する処理や数え上げ処理を行わずに、加算、減算、シフト演算のみの処理で済ませることにより、高速にブレンド比率を算出できる。

【0014】また本発明によれば、ブレンド比率を一旦蓄えるブレンド比率記憶手段を備えることにより、算出したブレンド比率を再利用でき、高速に多階調文字を出力できる。

【0015】さらに本発明によれば、ブレンド比率を一旦蓄える手段において、文字の存在領域を示すマスクパターンを、ブレンド比率とは別に蓄えることにより、ブレンド比率を管理しやすくなることができる。

【0016】

【作用】文字を出力するためには、文字のベクトルデータの読み込み、ブレンド比率算出、ブレンドの一連の処理が必要となる。

【0017】まずはじめに、文字のベクトルデータを読み込み、展開する。

【0018】次にブレンド比率を算出するために、水平線ごとに、展開した文字の輪郭線との交点を検出し、数値計算のみの処理で各ドットごとに文字部分の占める面積を近似してブレンド比率を算出することにより、高速

にブレンド比率を求めることができる。

【0019】そして、前記ブレンド比率算出手段により算出されたブレンド比率を用いて、ブレンド手段に基づき、文字色として指定された色と背景色とを混合して階調色を決定し、出力する。

【0020】その結果、従来のように出力するたびに必要とされていた各ドットごとのブレンド比率算出処理を高速に行なうことができるため、多階調文字の高速出力が可能となる。

【0021】さらにこの応用として、ブレンド比率算出手段で求めたブレンド比率をブレンド比率記憶手段に記憶して再利用すれば、背景色及び文字色が異なっても、同じサイズの同じ文字であれば、同様なブレンド比率算出処理をしなくて済む。

【0022】また、文字の存在領域を示すマスクパターンを、ブレンド比率とは別に蓄えることにより、ブレンド比率を管理しやすくなる。

【0023】

【実施例】以下、本発明の1実施例を説明する。

【0024】図1は本発明の1実施例を示すシステム構成のブロック図である。

【0025】11は文字コードを入力するキーボード、マウス等の入力手段であり、12は一連の処理を制御する制御装置であり、13はフォントデータを保存しておくフォントデータ記憶手段であり、14は文字と多図形との色合成比率を各ドットごとに計算するブレンド比率算出手段であり、15は前記ブレンド比率算出手段より算出した各ドットごとのブレンド比率を一旦蓄えるブレンド比率記憶手段であり、16は前記ブレンド比率記憶手段に蓄えておいたブレンド比率に基づいて実際に合成を行なうブレンド手段であり、17は前記ブレンド手段により生成された多階調文字を描画するフレームメモリであり、18はフレームメモリに描画された画像情報を出力する。表示装置、プリンタ等の出力手段である。

【0026】前記各部の動作について説明する。

【0027】まず、出力要求された文字が入力手段11より入力される。次に制御装置12により、要求された文字のデータをフォントデータ記憶手段13より読み込む。次にブレンド比率算出手段14では、まずフォントデータを展開し、要求された文字を構成するすべてのドットにおいて文字領域の占める割合をブレンド比率として求める。次にブレンド手段16により、文字を構成するすべてのドットに対して、各ドットごとに求めたブレンド比率に基づき、文字色及び背景色を混合して色合成を行ない、多階調文字を生成する。そしてフレームメモリ17ではブレンド手段16で生成した多階調文字を描画する。

【0028】ブレンド比率記憶手段15は、前記ブレンド比率算出手段14で算出したブレンド比率を一旦記憶するためのものである。

【0029】前記ブレンド比率算出手段14では、文字を構成する各ドットにおいて、文字領域の占める割合をブレンド比率として算出し、求めたブレンド比率を0%から100%に設定しておく。そして前記ブレンド手段16において、このブレンド比率に基づいて文字色として指定された色と背景色とを混合して階調色を求めれば、各ドットごとに文字領域の占める割合に応じた階調色を求めることができ、特に文字の輪郭部分や斜線などは背景色を考慮した自然な色合いとなり、斜線部分のギザギザが目立たない表示結果を得ることができる。

【0030】図9及び図12を用いてブレンド比率算出手段14の動作について説明する。図9はドットパターンの表示画面に対応するものであって、フォントデータ記憶手段13には、そのドットパターン内において例えばアルファベットの「N」の輪郭座標データを含めてその他の各種の文字の輪郭座標データが記憶されている。この輪郭座標データが文字ベクトルデータである。

【0031】まず、図9に示すように、ドットパターン上に読み込んだ文字ベクトルデータを重ね合わせた状態で、例えば図9の円内を取り出したものを図12とする。図12に示すように各ドットの文字領域がその各ドット内に占める面積割合（文字領域の占める面積÷1ドットの面積）を算出する。これがブレンド比率である。

【0032】すなわち、ドットd1でのブレンド比率は98%、ドットd2でのブレンド比率は75%、ドットd3でのブレンド比率は0%、ドットd4でのブレンド比率は70%、ドットd5でのブレンド比率は96%、ドットd6でのブレンド比率は24%、ドットd7でのブレンド比率は24%、ドットd8でのブレンド比率は98%、ドットd9でのブレンド比率は72%となる。

【0033】前記ブレンド比率算出手段14において、文字を構成する各ドットのブレンド比率を（文字領域の占める面積÷1ドットの面積）で算出すれば、各ドットごとに文字領域の占める割合を正確に反映させたブレンド比率を求めることができる。

【0034】図2は前記ブレンド比率算出手段14の1実施例を示す処理フローである。

【0035】まず、ステップST21において、文字のベクトルデータを読み込む。ステップST22では、文字のベクトルデータを展開し、文字の輪郭線を生成する。ステップST23では、1ドットを階調数だけ横に分割した単位で水平線を発生させ、水平線ごとに文字の輪郭線と水平線との交点を算出する。次にステップST24において、図4に示すように、ステップST23で算出した交点から文字領域を覆うような短冊を考える。この短冊の高さdyは分割数に依存しているため一律に決まる。また幅dxは交点x1、x2を使って、 $dx = x2 - x1$ より容易に求まる。ステップST25において、各短冊の面積を $dx \times dy$ により算出し、ドット内における各短冊の総面積を求める。ステップST26で

は（短冊の総面積÷1ドットの面積）よりブレンド比率を算出する。

【0036】本実施例によれば、ワークメモリ上に文字を描画せずにブレンド比率を算出できるので、ブレンド比率算出時間の短縮が可能となる。さらに、膨大なワークメモリを確保する必要もない。

【0037】また、本実施例において、1ドット全体が文字部分に覆われている場合、及び全く覆われていない場合には上記ブレンド比率を算出する処理を行なわなければ、文字の輪郭部分の処理のみで済むので高速に多階調文字を生成することができる。

【0038】次に、図5を用いて別の実施例を説明する。

【0039】ここでは、上述の実施例において短冊の高さdyが1になるように調整した場合を示す。

【0040】ステップST51において、文字のベクトルデータを読み込む。本実施例も前記実施例と同様に短冊を生成する。しかし本実施例の場合には、短冊の高さが1になるように、あらかじめステップST52において、ベクトル座標データを階調数倍に拡大する。階調数は通常2、4、16、64、256等の2のべき数で表わされる。そのため、各々1、2、4、6、8ビットシフトすることにより拡大する。次にステップST53において、拡大した文字のベクトルデータを展開し、文字の輪郭線を生成する。ステップST54では、1ドットごとに水平線を発生させ、文字の輪郭線と水平線との交点を算出する。ステップST55では、ステップST54で算出した交点より文字領域を覆うような短冊を生成する。この短冊の高さdyはあらかじめ1になるようにステップ52で調整してあるため、一律に1となる。また幅dxは交点x1、x2より求まる。ステップST56において、各短冊の面積を算出する。短冊の高さdyは1であるので短冊の面積は、すなわち短冊の幅となる。したがってドット内における各短冊の総面積は各短冊の幅の総和となる。ステップST57では、はじめにベクトル座標データを階調数倍にしておいたため、（短冊の総面積÷拡大した1ドットの面積）よりブレンド比率を算出する。このとき、拡大した1ドットの面積も前記説明したように2のべき数となる。そこで同様にしてビットシフトで除算を行なう。

【0041】この実施例によれば、あらかじめ座標データを階調数倍しておくため、短冊の面積及び総面積を求めるのに整数の加減算及びシフト演算のみの処理で済むので、計算を単純化することができ、計算時間の短縮が可能となる。

【0042】本実施例の場合にも、膨大なワークメモリを使用する必要はない。

【0043】これら2つの実施例は水平線ごとの処理であるため、文字サイズが大きくなっても処理時間が大幅に増加することはない。

【0044】また、この実施例の場合にも前記実施例と同様に、1ドット全体が文字部分に覆われている場合及び全く覆われていない場合に上記ブレンド比率を算出する処理を行なわないようにすれば、高速な多階調文字生成が可能となる。

【0045】次にこれら2つの実施例において、ブレンド比率とは別に文字の存在領域を示すマスクパターンを持つ場合について説明する。

【0046】例えば、文字を4階調で出力する場合には、もとの1ドットの状態は図10のうちのいずれかで表現できると考えられる。ここでは1ドットを4個のサブピクセルで表現し、各々のサブピクセルごとに文字領域とするかどうかの判定を行ない、1ドットにおける文字領域の占める割合を求めてブレンド比率とする。この場合、ドットd1のブレンド比率は0%、ドットd2のブレンド比率は25%、ドットd3のブレンド比率は50%、ドットd4のブレンド比率は75%、ドットd5のブレンド比率は100%となり、それぞれドットd1、ドットd21、ドットd31、ドットd41、ドットd51のように示され、5通りに表現されることになる。従って、この場合ブレンド比率を管理するために必要なビット数は4階調の場合3ビットとなる。同様にして16階調の場合5ビット、64階調の場合7ビット、256階調の場合9ビットとなる。

【0047】そこで、文字を構成する各ドットごとに文字の存在情報を管理するような1ビットのフラグを持たせる。そして例えば文字部分がドット上に少しでも存在している場合にはオン、全く存在していない場合にはオフとする。例えば図13(a)の「立」の場合には図13(b)のようなマスクパターンを持たせる。同様にして図10の場合には、ドットd1の文字存在情報フラグはオフ、ドットd2、ドットd3、ドットd4、ドットd5の文字存在情報フラグはオンとなる。

【0048】このように表現すると、4階調の場合には文字存在情報フラグがオフとなるのはブレンド比率0%のときのみとなり、オンのときは25%、50%、75%、100%の4通りのブレンド比率をとることになる。ブレンド比率とは別に文字の存在領域を示すマスクパターンを持たせることによりブレンド比率だけを管理するのに必要なビット数は2ビットとなる。同様に、16階調の場合には4ビット、64階調の場合には6ビット、256階調の場合には8ビット必要になる。

【0049】本実施例によれば、ブレンド比率を表現するためのビット数が階調数により決定できるため、必要最小限の管理情報量で済み、256階調の場合にもブレンド比率を高々8ビットで表現できるので、管理及びメモリ削減の面で役立つ。

【0050】さらに、マスクパターンを持たせておけば、フラグがオフの場合には、ブレンド比率算出処理を行なう必要はない。従って、ブレンド比率算出処理が必

要でないドットを見つけだすのにも便利である。

【0051】次に、図6を用いて、別の実施例を説明する。

【0052】本実施例は、前記実施例の応用として、一度算出したブレンド比率をブレンド比率記憶手段15に一時記憶しておき、再利用するものである。

【0053】まず、ステップST61において、文字コードを指定する。ステップST62では、要求された文字のブレンド比率がブレンド比率記憶手段15に既に記憶されているかどうか検索する。要求された文字を構成する各ドットごとのブレンド比率が既にブレンド比率記憶手段15に記憶されている場合には、ブレンド比率記憶手段15に記憶されたブレンド比率情報と、背景色及び文字色より、ステップST66で階調色を決定する。ステップST62において、文字のブレンド比率情報がブレンド比率記憶手段15に記憶されていない場合には、ステップST63でベクトル文字情報から文字を構成する各ドットごとのブレンド比率を算出する。

【0054】次にステップST64でブレンド比率記憶手段15に文字のブレンド比率情報を記憶する領域があるかどうかをチェックする。もし空き領域があれば、ステップST65でブレンド比率記憶手段に文字のブレンド比率情報を記憶する。空き領域がない場合には、ステップST68において既にブレンド比率記憶手段15に記憶している文字の中で、最も出力要求頻度の低い文字を検索し、ブレンド比率記憶手段15よりこの文字のブレンド比率情報を削除して新たな文字を記憶する領域を確保した後で、ステップST65でブレンド比率記憶手段15に文字のブレンド比率情報を記憶する。

【0055】次に、ステップST66で、ブレンド比率記憶手段15に記憶されている各ドットごとの文字領域の占める割合に応じて、背景色及び文字色を混合し、各ドットの階調色を決定する。そしてステップST67で、ブレンド手段16で作成した多階調文字をフレームメモリ17上に描画する。

【0056】上記ステップST68では出力要求頻度の低い文字を検索しているが、例えば、ブレンド比率記憶手段15に、記憶されている文字のブレンド比率情報の使用時刻を管理しておけば、使用時刻の最も古い文字を容易に検索できる。

【0057】つまり、ブレンド比率記憶手段15に文字を記憶する際、及びブレンド比率記憶手段15に記憶された文字を使用する際に、その使用時刻を記憶する。そして、文字削除を行なう際には、記憶してある使用時刻を調べ、その中で最近最も使われていない文字を削除する。

【0058】本実施例によれば、最初に文字出力を要求されたときだけ、ブレンド比率記憶手段15に算出したブレンド比率を記憶してしまえば、再び同じ文字を要求されたときには、それを再度利用することにより、同

じサイズで同じ文字のブレンド比率算出処理を冗長に行なわず、高速な多階調文字出力が可能となる。

【0059】さらに、文字のブレンド比率情報をブレンド比率記憶手段15に記憶できなくなった場合には、出力要求頻度の低い文字をブレンド比率記憶手段15より削除して、新たな文字を記憶することより、ブレンド比率記憶手段15に所望の文字の存在する確率（ヒット率）を高くすることができ、より高速に多階調文字を出力できる。

【0060】また、ブレンド比率は上記のような数値でなく、その他の表現方法であってもよいことはもちろんである。

【0061】さらに別の実施例として、上記ブレンド比率算出処理をソフトウェアで行なわず、LSIに内蔵する場合について図7、図11を用いて説明する。

【0062】図7はデータ処理装置の一例となるグラフィック表示装置の構成例を示したものである。システムは好ましくは単一の半導体基板に集積化された制御装置12、RAM122、好ましくは単一の半導体基板に集積化された文字処理装置123、フォントデータ124、フレームバッファ125、及びCRT等の出力手段17から成る。尚、図中、入力手段11は省略されているが、図1と同様に、制御装置12に接続されている。

【0063】制御装置12は、RAM122に記憶されたプログラムを実行処理し、システム全体を管理制御する。すなわち、文字処理において、文字処理装置123に対し、文字処理コマンドとパラメータ情報を転送し、文字処理装置123を起動する。

【0064】文字処理装置123は、加算器127、減算器128及びシフト器129を備えた装置であり、制御装置12からの指示に従い、フォントデータ124をアクセスし、あらかじめ定められた処理手順に従ってブレンド比率を算出し、フレームバッファ125上での処理を行なう。

【0065】本実施例では特に、フレームバッファ125には表示装置の各カラープレーンにおける各画素のブレンド比率を書き込むことにする。

【0066】文字処理装置123では、前記実施例で説明した方法でブレンド比率を算出する。図5で示した一連の処理について図18を用いて説明する。まず、フォントデータ124から座標データを読み込み、階調数倍に拡大する。このとき階調数は、2値、4階調、16階調、64階調、256階調と2のべき乗数(2のg乗)で表わせるため、全座標データをgビットシフトして拡大する。

【0067】次に、拡大した文字のベクトルデータを展開し、水平線ごとに文字の輪郭線と水平線との交点を算出し、ソートする。水平線ごとの交点算出は、文字の輪郭線の傾きから減算器だけを用いて容易に求めることができる。

【0068】算出した交点 $e_1$ 、 $e_2$  ( $e_1 > e_2$ ) より文字領域を覆う短冊を生成する。このとき短冊の幅は2交点の差 $A_{cc}$ で表わせる。ここではあらかじめ階調数倍に拡大してあるため、短冊の高さは1となる。従って、短冊の面積は短冊の幅 $A_{cc}$ で示されるため、結果的に短冊の面積は $A_{cc}$ で示され、減算器だけを用いて算出できる。

【0069】次に1ピクセルごとに短冊の総面積を求める。これは既に求めてある各短冊の面積の総和 (Blend) で求められるため、加算器だけを用いた処理となる。

【0070】そして各短冊の面積の総和 (Blend) より、背景の占める面積は(1ピクセルの面積 (Gray) - Blend) となる。そこでピクセルの色は文字色 (RGB)  $\times$  Blend + フレームメモリ (RGB)  $\times$  (Gray - Blend) で求められる。

【0071】ブレンド比率は(短冊の総面積 $\div$ 階調数)で求められる。階調数は前記説明したように2のg乗と表現できるため、gビットシフトするだけで容易に算出できる。

【0072】従って、ブレンド比率を求める処理は加算、減算、シフト演算のみで行なうことができる。そこで、加算器127、減算器128、及びシフト器129を備えた文字処理装置123をあらかじめ作成しておけば、ソフトウェアで処理するのに比べて、かなり高速にブレンド比率を算出することができる。

【0073】

【発明の効果】本発明によれば、以上説明したように構成されているので、以下に記載されるような効果がある。

【0074】各ドットごとにブレンド比率を算出する際に、数値計算のみの処理で行なうことより、高速に求めることができる。

【0075】また、数値計算のみの処理となるため、計算用のワークメモリを使わずにブレンド比率を求めることができる。

【0076】従って、高速にブレンド比率を算出できるため、メモリ節減及び多階調文字の高速出力が可能となる。

【0077】また、このブレンド比率算出処理をソフトウェアで行なわず、LSIに内蔵すれば、従来と比較してかなりの高速化が可能となる。

【0078】さらに、文字の存在領域を示すマスクパターンを、ブレンド比率とは別に蓄えることにより、ブレンド比率を管理しやすくなる。

【0079】また、最初に文字を要求されたときだけ、算出したブレンド比率をブレンド比率記憶手段に記憶してしまえば、再び同じ文字を要求されたときに、それを再利用できるため、高速な多階調文字出力が可能とな



る。

【0080】さらに、記憶されている文字のブレンド比率情報の使用時刻を管理しておけば、文字のブレンド比率情報を記憶する領域が無くなった場合に、出力要求頻度の低い文字を検索することができるため、常に使用頻度の高い文字を記憶しておくことができ、高速に多階調文字を出力することができる。

【図面の簡単な説明】

【図1】本発明のシステムを示すブロック図である。

【図2】本発明の1実施例の処理手順を示すフローチャートである。

【図3】従来から行なわれてきたブレンド比率算出処理を示す図である。

【図4】本発明のブレンド比率算出処理を示す図である。

【図5】本発明の別の実施例の処理手順を示すフローチャートである。

【図6】本発明の別の実施例の処理手順を示すフローチャートである。

【図7】本発明の別の実施例の構成図を示したものである。

【図8】多階調表示した場合の拡大図である。

【図9】ドットパターン上における文字ベクトルデータ図である。

【図10】本発明の手法でブレンド比率を求める際に仮想的に示されるドットの状態を示した図である。

【図11】本発明の別の実施例の処理手順を示すブロック図である。

【図12】ドット内における文字領域の面積割合を示す図である。

【図13】1実施例のマスクパターンを説明する図である。

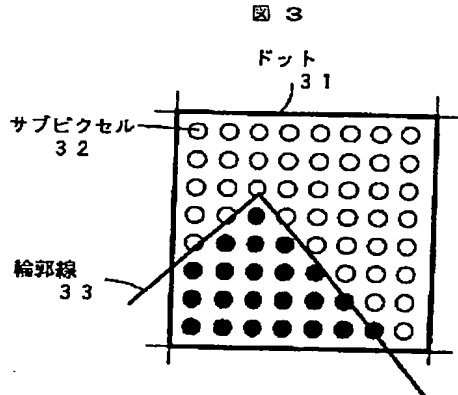
【図14】サブピクセルパターンのワークメモリを示したものである。

【図15】階調数の変化に伴い、ワークメモリがどのように増加するかを示した図である。

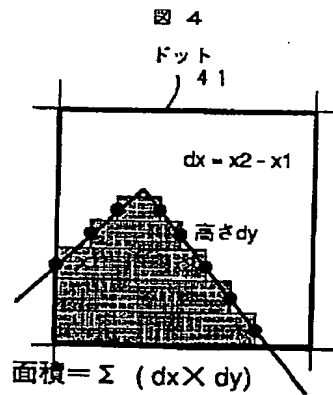
【符号の説明】

11…入力手段、12…制御装置、13…フォントデータ記憶手段、14…ブレンド比率算出手段、15…ブレンド比率記憶手段、16…ブレンド手段、17…フレームメモリ、122…RAM、123…文字処理装置、124…フォントデータ、125…フレームバッファ、31…ドット、32…サブピクセル、33…輪郭線、41…ドット。

【図3】

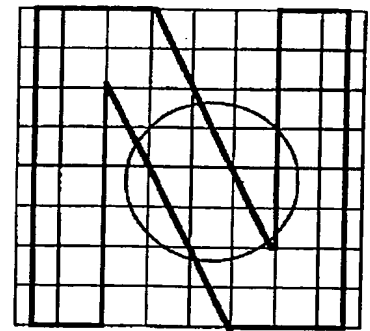


【図4】



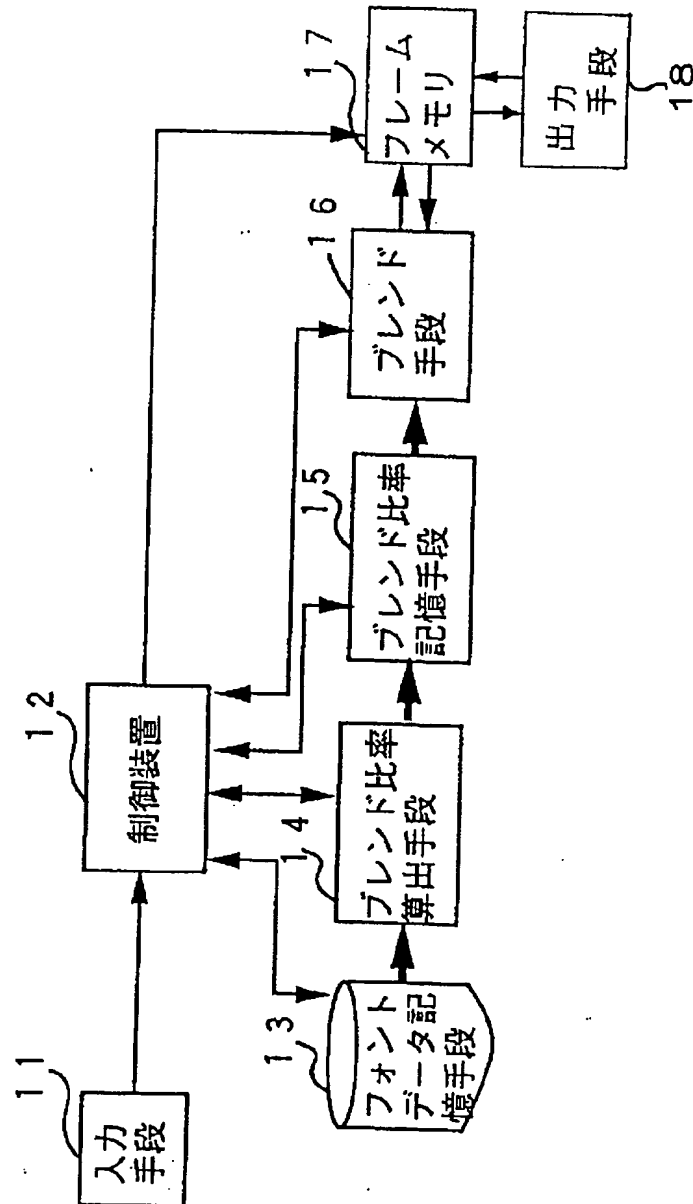
【図9】

図 9

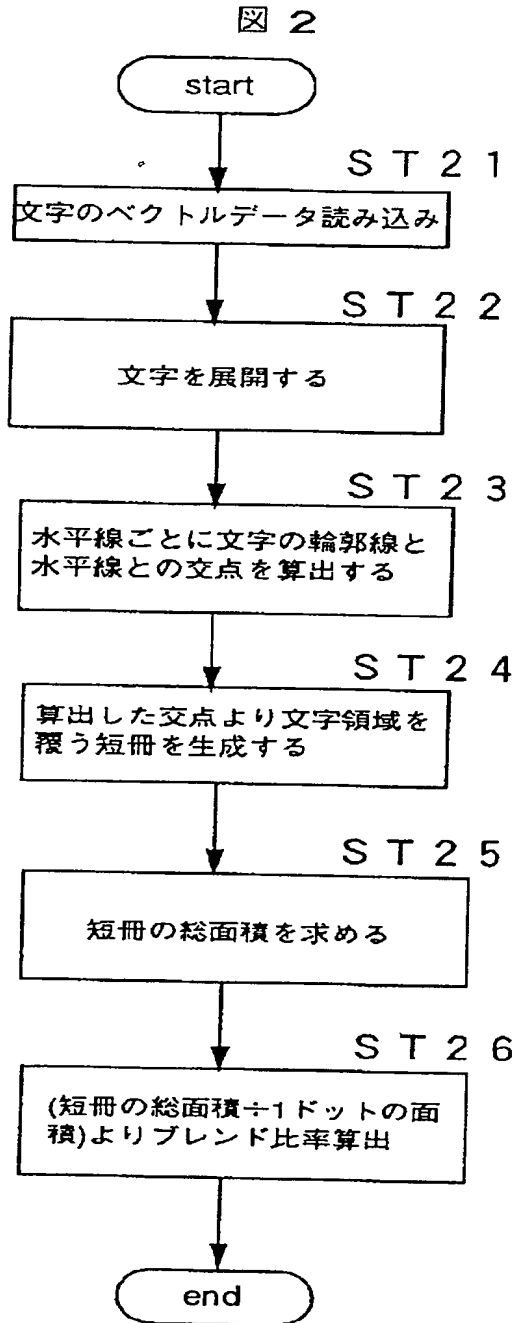


【図1】

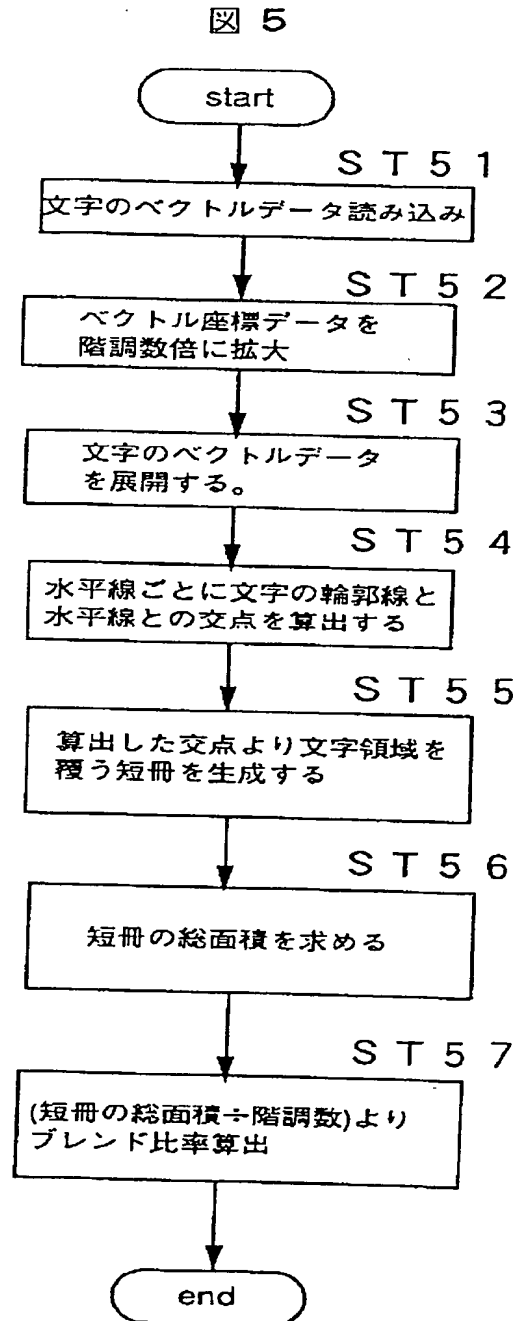
図 1



【図2】

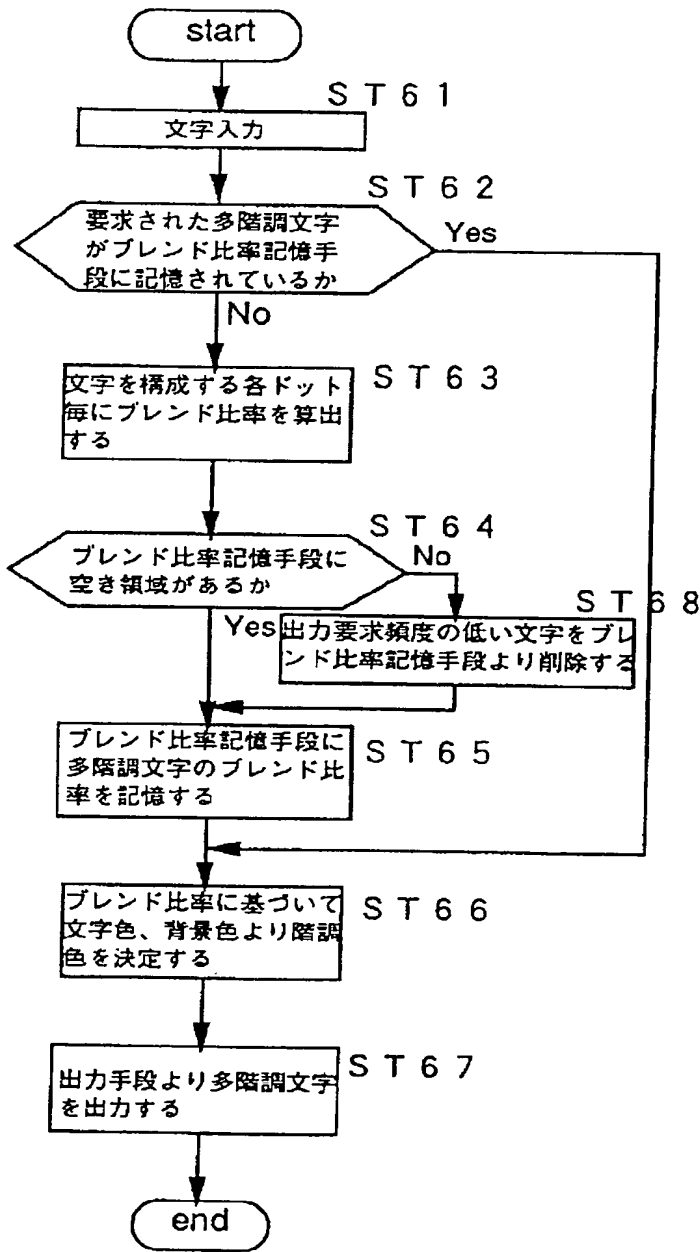


【図5】



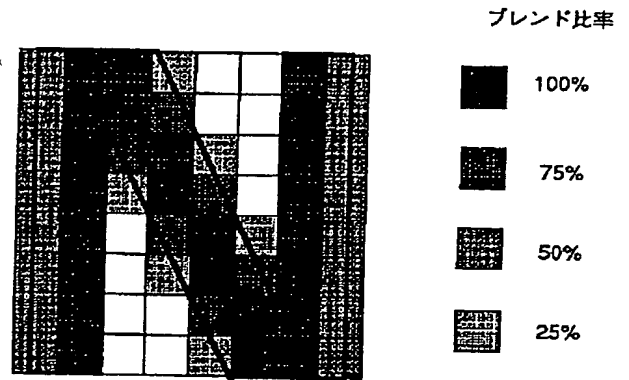
【図 6】

図 6



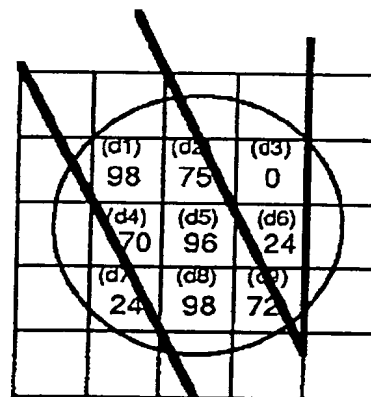
【図 8】

図 8



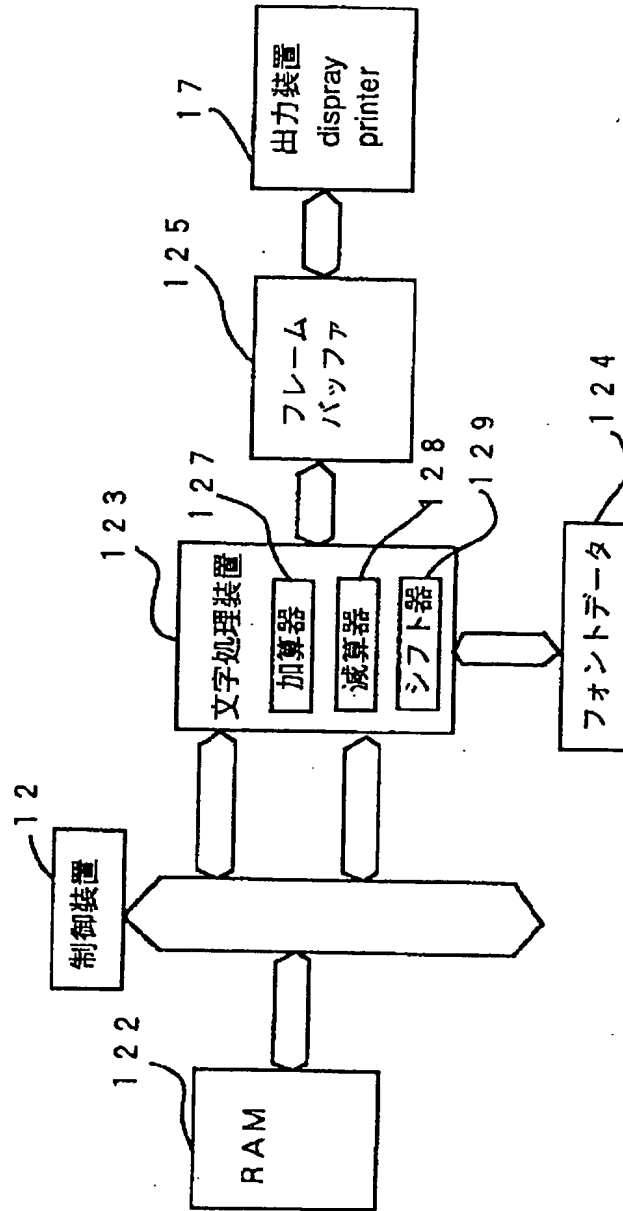
【図 12】

図 12

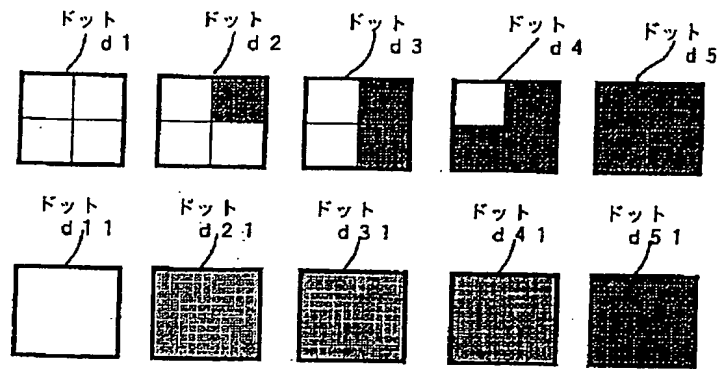


【図 7】

図 7



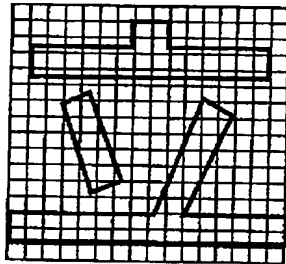
【図10】

図  
10

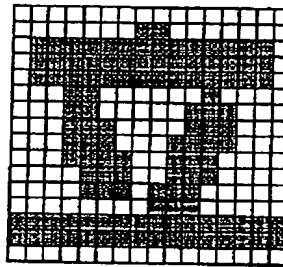
【図13】

図 13

(a)文字の輪郭線

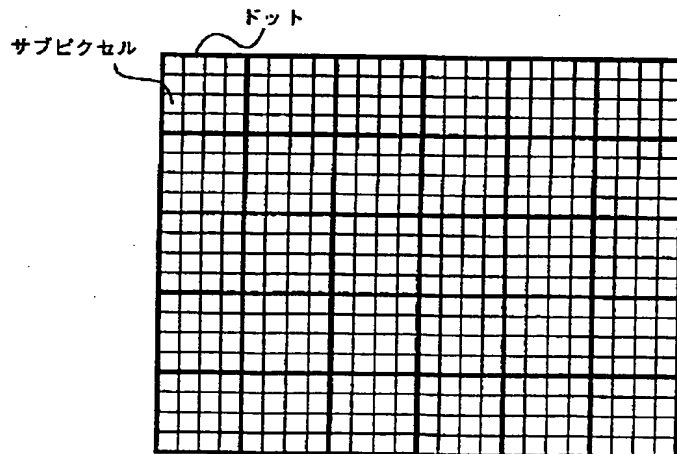


(b)マスクパターン



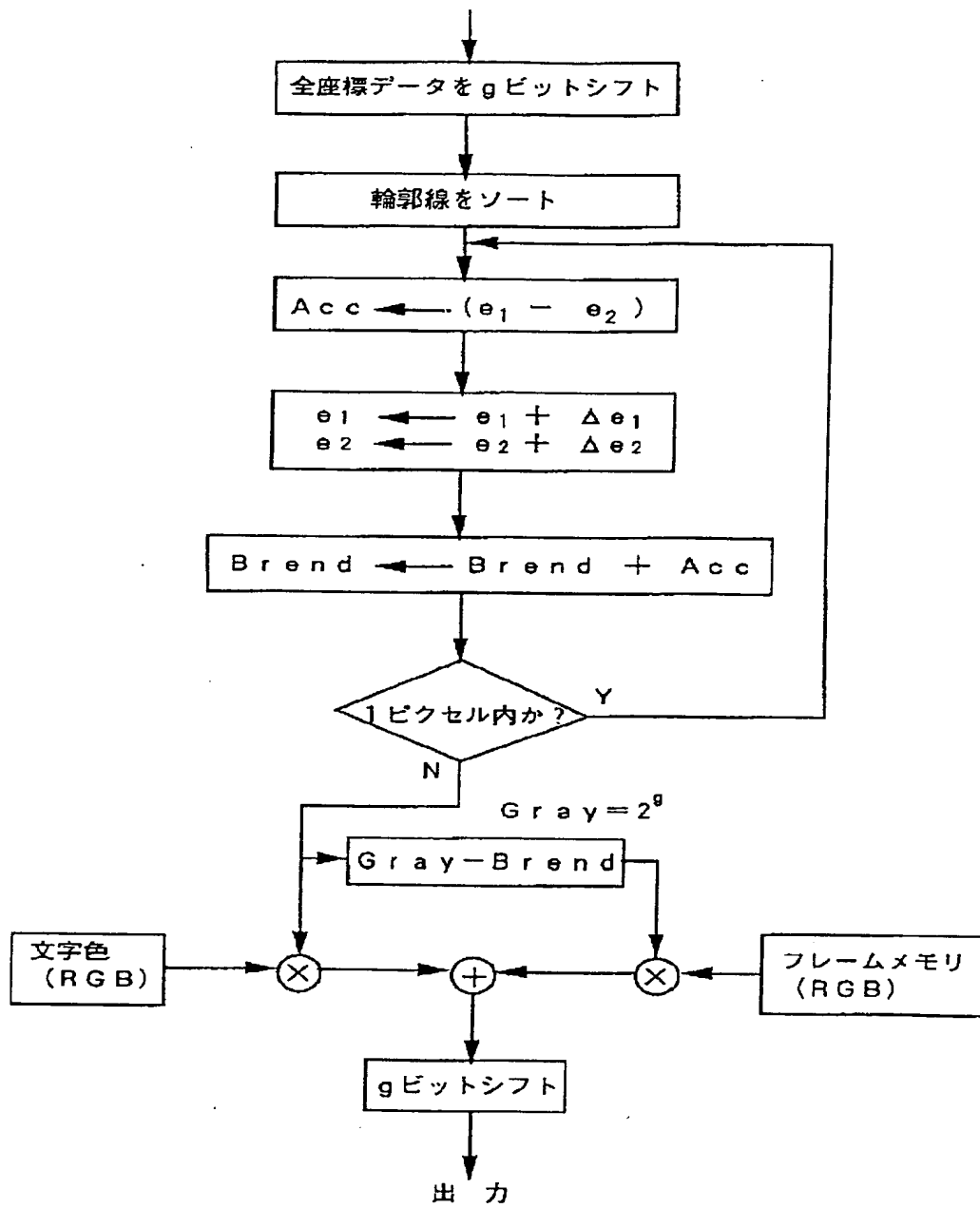
オン ■  
オフ □

【図14】

図  
14

【図11】

図 11



【図 1 5】

図 1 5

